

Forostar: A system for GIR

Simon Overell¹, João Magalhães¹ and Stefan R uger^{1,2}

Multimedia and Information Systems

¹Department of Computing, Imperial College London SW7 2AZ, UK

²Knowledge Media Institute, The Open University, Milton Keynes, MK7 6AA, UK
{simon.overell, j.magalhaes}@imperial.ac.uk and s.rueger@open.ac.uk

Abstract. We detail our methods for generating and applying co-occurrence models for the purpose of placename disambiguation. We explain in detail our use of co-occurrence models for placename disambiguation using a model generated from Wikipedia. The presented system is split into two stages: a batch text & geographic indexer and a real time query engine. Four alternative query constructions and six methods of generating a geographic index are compared. The paper concludes with a full description of future work and ways in which the system could be optimised.

1 Introduction

In this paper we detail *Forostar*, our GIR system designed to enter GeoCLEF 2006. We begin with a full outline of the system followed by our experimental runs. We aim to test the accuracy of our co-occurrence model and how the use of large scale co-occurrence models can aid the disambiguation of geographic entities. We conclude with an analysis of our results and future work.

We use a *rule-based* approach to annotate how placenames occur in Wikipedia (taking advantage of structure and meta-data). This annotated corpus is then applied as a co-occurrence model using a *data-driven* method to annotate the GeoCLEF data.

1.1 Discussion on ambiguity

GeoCLEF is an appropriate framework for developing different methods of placename disambiguation, however, evaluation is difficult as no ground truth exists for the corpus. The method we use for indexing locations is a *unique geographic index*, every occurrence of a placename is represented as a single polygon on the earth's surface in a spatial index. This allows for efficiently comparing the locations referred to in a query to documents in the corpus.

Despite there being minimal ambiguity in the GeoCLEF queries themselves, due to the use of a unique geographic index we would still expect to see an improvement in MAP as the accuracy of the index increases. This is because locations not necessarily occurring in the query but appearing in the footprint described by the query being incorrectly classified causing false negatives.

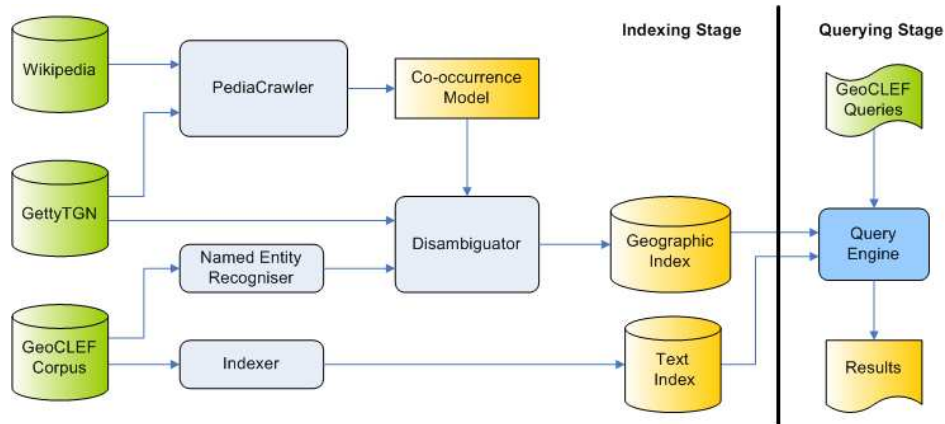


Fig. 1. System Design

2 The system

Forostar is split into two parts: the *indexing stage* and the *querying stage* (Figure 1). The indexing stage requires the corpus and some external resources to generate the geographic and text indexes (a *slow* task). The querying stage requires the generated indexes and the queries; it runs in *real time*.

The Indexing stage consists of four separate applications: *PediaCrawler* is first used to crawl the links in Wikipedia building a co-occurrence model of how placenames occur [9]; *Disambiguator* then applies the co-occurrence model to disambiguate the named entities extracted from the GeoCLEF corpus by the *Named Entity Recogniser* [2]. The disambiguated named entities form the geographic index. *Indexer* is used to build the text index.

The Querying stage consists of our *Query Engine*; this queries the text and geographic indexes separately, combining the results.

2.1 PediaCrawler

PediaCrawler is the application designed to build our co-occurrence model. Wikipedia articles and stubs are crawled¹; articles referring to placenames are mapped to locations in the Getty Thesaurus of Geographic Names (TGN). Our co-occurrence model takes the form of two database tables: the Mapping table, a mapping of Wikipedia articles to TGN unique identifiers; and the Occurrences table, links to articles believed to be places and the order in which they occur, no other information is used in the model.

PediaCrawler uses rule-based methods of disambiguation. It is made up of two parts, the *disambiguation framework* and a *method of disambiguation*. By using Wikipedia to build our co-occurrence model we hope to solve two problems:

¹ Our copy of Wikipedia was taken 3rd Dec 2005

the problem of *synonyms* (multiple placenames referring to a single location) is resolved by recording how multiple anchor texts point to the same page; and the problem of *polynoms* (a single placename referring to multiple locations) can be solved with our disambiguation system.

The disambiguation framework. The disambiguation framework is a simple framework to abstract the method of disambiguation from the document crawler. The framework is outlined as follows: *PediaCrawler* loads the Wikipedia articles to be crawled from a database, the links from each article are crawled in turn. For each link, if the article linked to has already been classified update the Occurrences table, otherwise, classify the article using the *Method of Disambiguation* specified and update both the Occurrences and Mapping tables.

The disambiguation method uses the following information to disambiguate a link: a set of candidate locations; a list of related placenames extracted from the metadata in the article pointed to by the link; and the text and title of the article pointed to by the link. Where the candidate locations are the set of matching locations found in the TGN by considering the link and title as placenames.

Our method of disambiguation. Based on the results observed by running a series of simple disambiguation methods on test data, we designed a disambiguation pipeline that could exploit the meta-data contained in Wikipedia and balance precision and recall (maximising the F_1 measure) [9].

Each disambiguation pipeline step is called in turn. A list of candidate locations is maintained for each article, an article is denoted as unambiguous when this list contains one or zero elements. Each method of disambiguation can act on the candidate locations list in the following ways: remove a candidate location; add a candidate location; remove all candidate locations (disambiguate as not-a-location); or remove all but one candidate locations (disambiguate as a location).

1. Disambiguate with templates – The template data in Wikipedia is highly formatted data contained in name-value pairs. The format of the templates is as follows: $\{\{template\ name\ | name_1 = value_1\ | \dots\ | name_i = value_i\}\}$. The template name is initially used for disambiguation, for example “Country” will indicate this page refers to a location of feature type nation or country. Templates are also used to identify non-places, for example if the template type is “Biographic” or “Taxonomic.” The name-value pairs within a template are also used for disambiguation, e.g. in the Coord template a latitude and longitude are provided which can be matched to the gazetteer.

2. Disambiguate with categories – The category information from Wikipedia contains softer information than the template information [7]; the purpose of categorising articles is to denote associations between them (rather than templates which are intended to display information in a uniform manner). Category tags can identify the country or continent of an article, or indicate an article is not referring to a location.

3. Disambiguate with co-referents – Often in articles describing a location, a parent location will be mentioned to provide reference (e.g. when describing

a town, mention the county or country). The first paragraph of the document is searched for names of containing locations. This method of disambiguation has been shown to have a high location-precision (articles correctly identified as locations) and grounding (articles correctly matched to unique identifiers), 87% and 95% respectively [9].

4. Disambiguate with text heuristics – Our heuristic method is based on the hypothesis: *When describing an important place, only places of equal or greater importance are used to provide appropriate reference.*

This hypothesis led to the implementation of the following disambiguation procedure: all the placenames are extracted from the first paragraph of the document; for each possible location of the ambiguous placename, sum the distance between the possible location and the extracted locations with a greater importance; classify as the location with the minimal sum.

2.2 Named entity recogniser

News articles have a large number of references to named entities that quickly place the article in context. The detection of references to all named entities is the problem addressed in this part of the system. The named entity recogniser receives as input the GeoCLEF news articles and outputs the named entities of each article.

Named entity recognition systems rely on lexicons and textual patterns either manually crafted or learnt from a training set of documents. We used the ESpotter named entity recognition system proposed by Zhu et al. [14]. Currently, ESpotter recognises people, organisations, locations, research areas, email addresses, telephone numbers, postal codes, and other proper names. First it infers the domain of the document (e.g. computer science, sports, politics) to adapt the lexicon and patterns for a more specialised named entity recognition which will result in a high precision and recall.

ESpotter uses a database to store the lexicon and textual pattern information; it can easily be customised to recognise any type of entity. The database we used is the one supplied by Zhu et al., we did not create a GeoCLEF specific database.

2.3 Indexer

The news article corpus was indexed with Apache Lucene 2.0 [1], later also used to search the article corpus. The information retrieval model used was the vector space model without term frequencies (binary term weight). This decision was due to the small size of each document that could cause a large bias for some terms. Terms are extracted from the news corpus in the following way: split words at punctuation characters (unless there is a number in the term); recognise email addresses and internet host names as one term; remove stop words; index a document by its extracted terms (lowercase) (see [1] for details).

2.4 Disambiguator

The *Disambiguator* builds a geographic index allowing results from a text search to be re-ordered based on location tags. The named entities tagged as placenames output by the *Named Entity Recogniser* are classified as locations based on their context in the co-occurrence model. This applies Yarowsky's "One sense per collocation" property [12].

The geographic index is stored in a Postgres database and indexed with an R-Tree to allow efficient processing of spatial queries [3]. In previous experiments we have shown the co-occurrence model to be accurate up to 80% [9], in this experiment we assume the geographic index to have an accuracy equal to or less than this.

Disambiguation methods: We compared three base-line methods of disambiguation to three more sophisticated methods:

No geographic index (NoGeo). In this method of disambiguation no geographic index is used. As with traditional IR methods, only the text parts of the query are executed. The motivation of this method is to measure to what extent the geographic index affects the results.

No disambiguation (NoDis). In this method no disambiguation is performed. Each placename occurrence is indexed multiple times, once for each matching location in the co-occurrence model. For example if "Africa" appears in a document it will be added to the geographic index twice: (Africa – Continent) and (Africa, Missouri, USA). This is analogous to a text only index where extra weight is given to geographic references and synonyms expanded. The motivation behind this method is to maximise the recall of the geographic index.

Most referred to (MR). For each placename occurrence, classify as the matching location that is referred to most often. The motivation behind this method is to provide a base-line to see if using sophisticated methods significantly improves results.

C-Index (CI). A co-occurrence index is assigned to every triplet of adjacently occurring placenames. The c-index represents the confidence with which a triple can be disambiguated as the most likely occurring locations. Triplets are disambiguated in descending order of c-index and disambiguated locations are propagated to neighbouring triplets. The motivation of this method is to see if only adjacent placenames are needed for disambiguation and if propagation of disambiguated placenames improves accuracy.

Decision List (DL). In this method of disambiguation we infer a decision list from the corpus and disambiguate each placename independently using the rule with the greatest confidence. The algorithm was originally suggested by Rivest [10], the version of the algorithm we use is described in detail by Yarowsky [13] and is similar to the work done by Smith and Mann [11]. The motivation of this method is to see if first order co-occurrence is all that is necessary for placename disambiguation.

Support Vector Machine (SVM). In the final disambiguation method we approach placename disambiguation as a vector space classification problem. In this problem the placenames can be considered as objects to be classified and

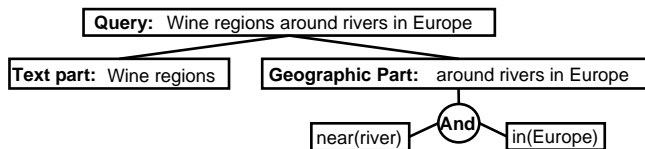


Fig. 2. Query Trees

the possible locations as classification classes. The chosen features were the placenames that co-occur with the placename being classified. The scalar values of the features are the inverse of their distance from the placename being classified, their sign is governed by whether they appear before or after the placename being classified. A separate feature space was built for each placename and linearly partitioned with a Support Vector Machine [6]. The motivation of this method is to see if multiple orders of co-occurrence can improve accuracy.

2.5 Query Engine

The *Query Engine* re-orders the results of the text queries produced by Lucene using the geographic queries.

The queries are manually split into a text component and a geographic component. The text query is handled normally by Lucene, the geographic query is manually split into a tree of conjunctions and disjunctions.

Executing a text query. Once the news articles have been indexed with Lucene, the query terms will be extracted in the same way as the document terms, a similarity measure is taken between the query’s terms and all indexed documents. The similarity function is given by the following expression:

$$\text{score}(q, d) = \frac{\sum_{t \in q} \text{tf}_t(d) \text{idf}^2(t) \text{norm}(d)}{\sqrt{\sum_{t \in q} \text{tf}_t^2(d)}}$$

where $\text{tf}_t(d)$ is the t term frequency for the given document d (in our case is 0 or 1), $\text{idf}(t)$ is the inverse document frequency of term t , and $\text{norm}(d)$ is a normalisation constant given by the total number of terms in document d . See the [1] for details.

The query tree. The query trees are constructed by hand. The nodes of the tree are either conjunctions or disjunctions while the leaves of the tree are spatial-relation, location pairs (see Figure 2).

Executing a query. The documents that match both the geographic and the text query are returned first (as ranked by Lucene). This is followed by the documents that hit just the text query. The tail of the results is filled with random documents.

3 Experimental runs & Results

We have executed 24 runs with the GeoCLEF 2006 data: all mono-lingual, manually constructed English queries on an English corpus. The queries constructed

with: the query topic and description (TD); the query topic, description and narrative (TDN); the text part containing no geographic entities (Tx); and text part containing geographic entities (GTx). We have four different query constructions: TD-Tx, TDN-Tx, TD-GTx and TDN-GTx.

The query constructions were tabulated against the six disambiguation methods. As far as was possible we attempted to add no world knowledge, the query trees produced resemble trees that could be generated with a query parser.

Our [non-base-line] runs appeared between the 25% and the 75% quartiles for mean average precision with most results around the median (results presented here were not included in the quartile calculations). We applied two sets of significance testing to our per query results. The Friedman test for multiple treatments of a series is a non-parametric test that can show differences across multiple treatments. The Wilcoxon signed-rank test is a non-parametric test that shows if two independent treatments have a significant difference and if there is a difference, which treatment is significantly better [5].

Mean Average Precision							Distribution	
	NoGeo	NoDis	MR	CI	DL	SVM	Worst	4%
TD-Tx	8.3%	17.1%	16.3%	16.6%	16.7%	16.3%	Q1	15.6%
TDN-Tx	8.4%	17.9%	19.4%	17.9%	19.4%	19.4%	Median	21.6%
TD-GTx	18.6%	22.9%	22.5%	21.3%	22.6%	22.5%	Q3	24.6%
TDN-GTx	21.7%	21.6%	22.2%	18.8%	22.2%	22.2%	Best	32.2%

The runs consisting of Title, Description and Narrative (TDN) statistically significantly out performed the respective runs consisting of only Title and Description (TD) when no geographic entities were contained in the text part of the query (Tx). The runs with geographic entities in the text part of the query (GTx) statistically significantly out performed the respective runs without geographic entities in the text part of the query (Tx). There was no statistical significance between the different geographic index runs (NoDis, MR, CI, DL and SVM). The runs using a geographic index were statically significantly better than the run without (NoGeo).

4 Conclusions and Future Work

We can conclude that the combination of geographic and text indexes generally improves geographic queries. To maximise MAP geographic phrases should be included when querying *both* the geographic index and the text index. This conclusion is consistent with previous experiments on the GeoCLEF data [4, 8]. We can also conclude that our experiments show no significant difference in the MAP achieved from using any of our methods for generating a geographic index. The inclusion of the narrative information increased MAP only when there was no geographic information contained in the text part of the query or no geographic index was used. This is because the narrative of a query specifies the geographic phrase in greater detail mainly adding information already contained in the geographic index.

With respect to our objectives we can conclude that the co-occurrence model accuracy agrees with the previous experiments conducted in [9] and that co-occurrence models are a suitable method of placename disambiguation. By increasing the size & accuracy of the co-occurrence model, increasing the number of queries and improving how the different indexes are combined, we believe in future experiments the improvement produced by disambiguating placenames will increase.

Lucene was applied in the default configuration and the text part of the queries were not altered in any way. We plan to experiment with suitable query weights for Lucene and try alternative configurations of the index. Ultimately we would like to combine the geographic and text indexes so that they can be searched and applied simultaneously. We also plan to implement a query parser to allow the queries to automatically be parsed into query trees; this would require a level of natural language processing.

References

1. Apache Lucene Project. <http://lucene.apache.org/java/docs/>. accessed 01 November 2006, 2006.
2. P. Clough, M. Sanderson, and H. Joho. Extraction of semantic annotations from textual web pages. Technical report, University of Sheffield, 2004.
3. A. Guttman. R-Trees, A dynamic index structure for spatial searching. In *SIGMOD International Conference on Management of Data*, 1984.
4. C. Hauff, D. Trieschnigg, and H. Rode. University of Twente at geoCLEF 2006: geofiltered document retrieval. In *Working Notes for GeoCLEF*, 2006.
5. D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *16th Annual ACM SIGIR*, 1993.
6. T. Joachims. *Advances in Kernel Methods – Support Vector Learning*. 1999.
7. D. Kinzler. Wikisense – Mining the Wiki. In *Wikimania '05*, 2005.
8. B. Martins, N. Cardoso, M. Chaves, L. Andrade, and M. Silva. The university of lisbon at GeoCLEF 2006. In *Working Notes for GeoCLEF*, 2006.
9. S. Overell and S. Rüger. Identifying and grounding descriptions of places. In *SIGIR Workshop on Geographic Information Retrieval*, 2006.
10. R. Rivest. Learning decision lists. *Machine Learning*, 1987.
11. D. Smith and G. Mann. Bootstrapping toponym classifiers. In *HLT-NAACL Workshop on Analysis of Geographic References*, 2003.
12. D. Yarowsky. One sense per collocation. In *ARPA HLT Workshop*, 1993.
13. D. Yarowsky. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *32nd Annual Meeting of the ACL*, 1994.
14. J. Zhu, V. Uren, and E. Motta. ESpotter: Adaptive named entity recognition for web browsing. In *Professional Knowledge Management Conference*, 2005.